The WIYN Arcon-IRAF Interface: A User's Guide

Dave Silva, Diana Kennedy, Phil Massey, and Taft Armandroff

Guide Beta Version 0.6: 2 August 1995 (drs)

Next Version ETA: 1 September 1995

Please send comments/corrections/typos to Dave Silva (silva@noao.edu) for incorporation in next version.

1.0 Introduction

This is a user's guide for the IRAF based user interface to the **Harcon** and **Arcon** CCD controllers, hereafter referred to generically as the **Arcon** controllers. These controllers are currently in use at WIYN to control the Imager CCD (a. k.a. S2KB) and the Bench Spectrograph CCD (a.k.a. T2KC). The WIYN Arcon-IRAF interface is also used to control the Imager **Filter Shutter Assembly (FSA)** (a.k.a. the Imager "filter wheel").

This guide is maintained on-line as a HTML document. If you have a hard copy of this guide, it is strongly suggested that you verify that the hard copy version number and revision date matches the on-line version. The on-line URL is TBD.

This guide assumes a familiarity with basic IRAF usage and functionality.

Important: This manual covers operation of WIYN Imager CCD and the WIYN MOS Bench Spectrograph CCD.

1.1 Supplementary Documentation

The observer might find the following documents useful:

- For an overview of IRAF, we suggest you read *A Beginner's Guide to Using IRAF* by Jeannette Barnes.
- For a basic description of CCD reductions, we suggest *A User's Guide to CCD Reductions with IRAF* by Phil Massey.
- For the definitive overview of MOS/Hydra, we suggest the soon-to-be-revised *Hydra User's Manual* by Sam Barden & Taft Armandroff.

All three of this supplementary documents can be obtained by contacting Jeannette Barnes (jbarnes@noao.edu).

2.0 Taking Data

2.1 The Only Command You Really Need To Know: *observe*

2.2.1 The WIYN Imager

All data taking can be done by using a single command: **observe**. The observer will be prompted for all the information necessary for controlling the exposure. An Imager *observe* dialogue looks like this (with user input in **bold**):

```
cl> observe

Exposure type (|zero|dark|object|dflat|sflat|focus) (zero): object
Number of exposures to take (1:) (1):
Exposure time (0.:) (0.): 1000
Title of picture (bias 16/07/93): HH80/81 [SII]
Filter in wheel one (U): I

Image obj107 written to disk

Observation finished...

[Running post processing command: postproc obj107]
```

Information requested includes:

- **exposure type:** can be "zero" (sometimes referred to as bias), "dark", "object", "dflat", "sflat" or "focus". Note that when selecting from a list of options like this you may enter any unique abbreviation. Focus sequences are **not** supported in this release of the software.
- **number of exposures to take:** a sequence of this number of pictures, all having the same parameters, will be taken.
- **exposure time:** is in seconds, and will not be requested in the case of exposures of type "zero".
- **picture title** will be included as the title in the IRAF image header.
- **filter in wheel one/two/three:** the required filter. Note that filters can be specified by name, rather than by their position in the filter wheel; the way this translation is set up will be described in section 3.3. The filter wheels will be moved to the selected positions before continuing. The filter used will also be noted in the image header. This information is not requested for exposures of types "zero" and "dark" for which the filter is irrelevant.

  *Note that the FSA only holds one (1) filter wheel at a time, although three filter wheels exist. The FSA is "smart" enough to sense what filter wheel is physically in place and will warn the observer if the wrong filter wheel is requested. However, if the wrong wheel is requested during an **observe** setup, the error described in the next paragraph will occur.*

  ***Warning:** If the filter fails to move to the correct position a warning message will be printed and the current **observe** command will abort.*

Note that in each parameter query you will be supplied with a default value, which you can accept by simply hitting <RETURN&GT;; these values are just the previous entries.

If you make a mistake, or change your mind, you can abort the command during the parameter entry stage by typing **CTRL-C**; having done so you should always enter the command **flpr**, to clean up the hidden mess. Once the exposure has started it can be terminated using the **abort** command (see Note that the **observe** command terminates *as soon as the exposure starts* and you can enter other commands in the IRAF Acquisition window. *Additional Arcon commands entered at this point can inadvertantly modify the current, active exposure.* This "feature" is a bug which will be corrected in later release of the software. In the meantime, *observers are urged **not** to issue additional Arcon commands while an exposure is on-going.* In addition, we suggest that all data testing and reduction commands be issued in the IRAF Data Reduction window, not the IRAF Data Acquisition window.

When the exposure finishes the CCD will be read out. The first line in the status window will change to "READING" and the "buffers read" counter will indicate the number of buffers of data successfully transferred to the Sun. The data is initially written in the controllers internal format to a spool file on /tmp, but it is automatically converted into an IRAF image within a few seconds of the exposure finishing. The message "Image ...... written to disk" appears as soon as this process is complete and shows you the name of the new IRAF image. This name is derived from the **exposure type** by appending a running number (controlled by the **obspars** parameter set described in section 3.2) which is automatically incremented after each exposure. The image header will be in the current directory (when the **observe** command completes) and the pixel file will be located in your **imdir**. A temporary file .T*image name*.imh will be in your current directory until the complete image is available. Then it is renamed to *image name*.imh.

If you requested that **observe** take only a single exposure, the message "observation finished ....." will appear in the IRAF interface window as soon as the readout is complete; things are then ready for you to start another exposure. If, instead, you requested a sequence of several pictures, the next exposure will start automatically. You may immediately examine or process the resulting image even though the sequence is not complete. Note that the "pictures remaining" counter in the status window shows how many exposures remain in the sequence. Once the final picture has been read out the message "sequence finished ......" will appear in the IRAF interface window. Should you miss the end of sequence or end of exposure message, note that the CCD is idle and things are ready for you to initiate new exposures, whenever the top line of the status display reads "CONTINUOUSLY_ERASING".

2.1.2. The MOS Bench Spectrograph

The MOS Bench Spectrograph CCD Arcon system behaves in nearly the identical way to the Imager CCD Arcon system with two notable exceptions:

- The Bench Spectrograph Arcon system has no interface to any Bench Spectrograph hardware expect the actual CCD system. All Bench hardware is controlled from the Bench Spectrograph Graphical User Interface (the "bsgui").
- The Bench Spectrograph Arcon system supports one additional image type, **comp**, for comparison spectra exposures.

Appendix C Section 2).

2.3 Other Commands For Taking Data

In addition to observe, there are specific commands to take one or more pictures of each type:

- **dark** - Take one or more exposures of type *dark*
- **dflat** - Take one or more exposures of type *domeflat*
- **object** - Take one or more exposures of type *object*
- **sflat** - Take one or more exposures of type *skyflat*
- **zero** - Take one or more exposures of type *bias*
- **comp** - Take one or more exposures of type *comp*. *Only supported by the Bench Arcon system.*

Except, of course, for the exposure type, these commands take the same parameters (and prompt for them in the same order) as does **observe**. Apart from saving you entering that one extra parameter, use of these commands allows one to set default parameter values, and also select which parameters are prompted for according to picture type.

Another useful command is:

- **more** - Take one or more exposures exactly like the previous exposure.

If you type

```
cl> more
```

you will not be prompted for the number of exposures (as one might expect) but rather a single exposure will be taken (which more often than not is what you actually wanted to do). However, if you type

```
cl> more 10
```

ten more exposures will be acquired. *Note to ICE users: **mores** does not work.*

A very useful task when you want to take Imager calibration exposures, or exposures of the same object, in various filters is

- **doobs**

For example, to take sequences of 10 dome flats each in R (60s exposures), U (10s), V (5s) and B (5s), type:

```
cl> doobs
Exposure type (|object|dflat|sflat|): dflat
Number of exposures to take in each filter (1:) (1): 10
wheel one, two or three (one): three
list of filters in wheel3: R, U, V, B
List of exposure times: 60,10,5

The following pictures will be taken:

Pictures           Filter3          Exposure

100  109            R                 60
110  119            U                 10
120  129            V                  5
130  139            B                  5

Title for pictures: Dome flats 21 June 95
```

3.0 The Parameter Files

The **observe** command (and its derivatives **dark**, **dflat**, etc.) have been purposely kept simple. Additional flexibility can be achieved by modifying the options listed in three key parameter sets (psets). Many observers will only modify these psets once, at the beginning of their run. These parameters sets are:

- **detpars** - This pset controls the fundamentals of how the CCD is read out, i.e., binning, gain, regions of interest, etc.
- **obspars** - This pset controls the behavior of of **observe** (and its derivatives).
- **wheel1,wheel2,wheel3** - This 3-in-1 pset contains Imager FSA filter information. *There is no equivalent pset for the Bench Spectrograph.*

You should review these parameter files at the beginning of your run, and modify them as necessary. Many observers will never modify them again.

These parameter files can be listed by using the **lpar** command, e.g.,

```
cl> lpar obspars
```

and may be edited using the parameter editor, **epar**, or by simply typing the name of the parameter set e.g.,

`cl> epar obspars` -OR-

```
cl> obspars
```

To change a value, in either case, move the cursor up and down with the arrow keys until you are on the correct line and then simply type the new value followed by &LT;RETURN&GT;. When done editing the parameter file type **:q**. On the WIYN workstations, **cntrl-Z** works as well.

3.1 detpars

The detector parameters are set in the pset detparm.par. The task **detpars** will automatically bring up the detparm.par file for editing and download the appropriate binary executables to the Arcon controllers (a.k.a. "download the waveforms"). The detector parameters are:

```
                  IRAF

        (gain = 4)                   Gain setting
    (preflash = 0.)                  Preflash time (seconds)
        (xsum = 1)                   pixels summed in X direction
        (ysum = 1)                   pixels summed in Y direction
      (xstart = 1)                   Start of ROI in X
      (ystart = 1)                   Start of ROI in Y
       (xsize = 2048)                Size of ROI in X
       (ysize = 2048)                Size of ROI in Y
      (extend = "separate")          Method of extending ROI to include overscan
      (xskip1 = 0)                   X pixels to skip at start of overscan
      (xskip2 = 0)                   X pixels to skip at end   of overscan
      (xtrim1 = 0)                   X pixels to trim at start of data
      (xtrim2 = 5)                   X pixels to trim at end   of data
      (ytrim1 = 2)                   Y pixels to trim at start of data
      (ytrim2 = 4)                   Y pixels to trim at end   of data
  (amplifiers = "lr")                Readout amplifiers to be used
     (pixsize = 21.)                 Pixel size in microns
    (nxpixels = 2048)                Detector size in X
    (nypixels = 2048)                Detector size in Y
   (noverscan = 32)                  Number of overscan pixels
     (detname = "S2KB")              Detector identification
        (mode = "ql")
```

Observers will wish to examine, and perhaps modify, the following parameters:

- **Gain** - This parameter is a CCD gain *index* which is discussed further in Section 3.1.1.
- **Xsum** - This parameter controls how many input/on-chip pixels are summed in the X direction to form one output pixel in X direction.
- **Ysum** - This parameter controls how many input/on-chip pixels are summed in the Y direction to form one output pixel in the Y direction.

The next four parameters establish the CCD Region of Interest (ROI) to be read-out.

- **Xstart** - This parameter sets the first column to be read-out.
- **Ystart** - This parameter sets the first row to be read-out.
- **Xsize** - This parameter sets the number of columns to read-out.
- **Ysize** - This parameter sets the number of rows to be read-out.

The ROI parameters shown in the example above correspond to reading out the entire CCD area.

***MOS Bench Spectrograph note:** there is no reason to ever readout less than the entire CCD during normal MOS Bench Spectrograph observations as it is beneficial to record the area beyond the spectra to monitor scattered light levels. This extended area is ultimately used to correct for scattered light (see the Hydra User's Manual for more details).*

The **preflash** field is not relevant to WIYN applications. The **extend**, **xskip1**, **xskip2**, **xtrim1**, **xtrim2**, **ytrim1**, **ytrim2**, and **amplifiers** should be left set to their default values.

The fields **pixsize**, **nxpixels**, **nypixels**, **noverscan**, and **detname** are only informational. They can not (and should not) be changed.

The messages you will see when the waveforms are downloaded will look very similar to these:

```
   Preflash time changed from 0.000000 to 0.000000

   There are no regions of interest defined.

   There is one region of interest defined:

ROI   Lower Left Corner      # of Pixels      Upper Right Corner

 #        x      y            x      y             x      y
---     ----- -----          -----  -----          ----- -----

 1        1     1            2048   2048           2048  2048

*** Regenerating waveforms ***

/xp/run/waveforms/Harcon1.9

*** Recompiling waveforms ***

/xp/run/waveforms/Harcon1.9
csh /xp/run/macro/wdl s2kb
WDL revision 2.18

Binning factor 1:  Pixel readtime =    15 us.

Binning factor 2:  Pixel readtime =    18 us.

Binning factor 3:  Pixel readtime =    22 us.

Binning factor 4:  Pixel readtime =    26 us.
cp s2kb.nex /xp/run/macro/Harcon+Stis2048+Observe.nex

*** Suspending the sequencer ***

*** Reloading new waveforms ***
```

3.1.1 Setting the Gain

It is possible to change the slope time of the "Integrater" - usually erroneously referred to as the gain - which determines the number of e-/ADU, the readout noise, and the readout time. This can be changed by entering **detpars** and changing the gain entry to one of the possible options.

The recommended WIYN CCD gain values are:

- Imager (S2KB): 2.8 electrons/ADU (gain index = 3)
- MOS Bench Spectrograph (T2KC): 1.7 electrons/ADU (gain index = 4)

*For most Imager projects and **all** MOS projects, we strongly urge the observer to adopt the above gain values.*

Nevertheless, the observer can determine the available gain options and current settings using:

- **ccdinfo**.

For the Imager CCD S2KB, **ccdinfo** produces the following output

```
cl> ccdinfo


        (gain = 4)                   Gain setting
    (preflash = 0.)                  Preflash time (seconds)
        (xsum = 1)                   pixels summed in X direction
        (ysum = 1)                   pixels summed in Y direction
      (xstart = 1)                   Start of ROI in X
      (ystart = 1)                   Start of ROI in Y
       (xsize = 2048)                Size of ROI in X
       (ysize = 2048)                Size of ROI in Y
      (extend = "separate")          Method of extending ROI to include overscan
      (xskip1 = 0)                   X pixels to skip at start of overscan
      (xskip2 = 0)                   X pixels to skip at end   of overscan
      (xtrim1 = 0)                   X pixels to trim at start of data
      (xtrim2 = 5)                   X pixels to trim at end   of data
      (ytrim1 = 2)                   Y pixels to trim at start of data
      (ytrim2 = 4)                   Y pixels to trim at end   of data
  (amplifiers = "lr")                Readout amplifiers to be used
     (pixsize = 21.)                 Pixel size in microns
    (nxpixels = 2048)                Detector size in X
    (nypixels = 2048)                Detector size in Y
   (noverscan = 32)                  Number of overscan pixels
     (detname = "S2KB")              Detector identification
        (mode = "ql")


                      *** Table of gain values ***

       Detector = S2KB,  2048 x 2048, 21 um pixels


     dcsT GAIN       Read_Noise       Gain               Readout Time
     (us)  #            (e-)        (e-/ADU)                (sec)
     ---  ---       -----------   -----------           ------------
1:    3    1            14            7.5                    88
2:    5    2             9            4.3                   104
3:    8    3             8            2.8                   129
4:   15    4             7            1.5                   185

       Full Well and Linearity to 200K e-
       ADC Saturates before FW at Gain #'s 3-4

     *** Select gain setting from the first column

     *** The current gain setting is 4
cl>
```

When you enter **detpars**, you can edit the gain, and when you enter **:q** it will download the waveforms and change the gain. To see if it has changed, do another **ccdinfo**.

3.2 obspars

The parameter file **obspars** contains four distinct groups of parameters as shown below:

```
       I R A F

     Image Reduction and Analysis Facility

 PACKAGE = fsa

 TASK = obspars

 ccdtype = zero         Exposure type
 npics = 1              Number of exposures to take
 picture = 1            Picture number of first exposure
 exposure = 0.          Exposure time
 title = Test frame     Title of picture
 (autopic = yes)        Generate picture number automatically ?

    # POST PROCESSING PARAMETERS

 (postpic = display $image) Post processing command for single picture
 (postseq =  )          Post processing command for sequences
 (restart = no)         Restart the server on every command?


    # SELECTING FILTER FOR EACH EXPOSURE

 (setfilt = one)        Query and set filters?
 (filtype = instrument) Type of filters to use



    # SETTING FOCUS FOR EACH EXPOSURE

 (setfocus = yes)       Query and set focus?
 (foctype = telescope)  Type of focus to use
 (tempera = 0.)         Telescope temperature
 (basefoc = INDEF)      Focus base value
 (reftemp = 0.)         Telescope temperature for base focus value
 (tfrcoef =  )          Coefficients of Temperature Focus relationship



   # PARAMETERS FOR FOCUS EXPOSURES

 nfexpo = 7             Number of focus exposures
 (refis =  middle)      Reference is first, middle or last exposure?
 freferen = 5000        Focus value
 fdelta = 30            Focus increment
 (focmode = auto)       Focus mode
 (shtype = detector)    Shift type
 (f_stepsize = 0.)      Focus stepsize in arc seconds
 (f_direction = north)  Focus direction for telescope
 (fnrows = 30)          Focus number of rows to shift
 (mode = ql)
 ($nargs = 0)
```

The first group of parameters are used for all exposures. It is not necessary to set the values of most of these, since they are prompted for as needed. The values appearing in **obspars** are simply the values entered the last time **observe** was run. The parameter **autopicnum** determines whether **observe** will prompt you for the running picture number, **picture**, which forms part of the name of your images on disk. The value of **picture** is always incremented after each exposure. If **autopicnum**=yes (the default) the automatically derived value will always be used and you will not be prompted. If **autopicnum**=no you will be prompted for a new value of **picture** for every exposure, the automatically derived value being supplied as the default. In either case you can reset the sequence by just changing the value of **picture** in obspars. ***Warning:** Note that **picture** will get out of step if you **abort** an exposure or sequence; the value used will be the one which would have been appropriate if the exposure or sequence had completed normally.*

The second group of parameters controls what task or script is run for postprocessing. One can chose a different script to be executed after the entire sequence from the one that is run after each exposure. The restart option determines whether a new job is created after each exposure or not. If **restart**=yes then a separate process will be created. This can cause too many processes within the **IRAF** environoment. It can be controlled by using a task call **postkill**. If **restart**=no, then you must not change the postprocessing scripts and you must remain in the current directory while taking data. If you want to change either of these, you must restart Arcon.

The third group of parameters controls the positioning of the Imager FSA filter wheel at the start of each exposure. ***For the Imager FSA*, setfilt** can be "none" not to move any wheel, "one" to move only wheel one, "two" to move only wheel two, or three to move only the third wheel. You will be prompted for the filter position when appropriate and the value you enter will be saved in the image header, *but you must wait several seconds for the filter to engage.* Remember that only one wheel is physically mounted in the FSA at any one time.

For the MOS Bench Spectrograph*, setfilt *should be set to "none".

Note that the filter position is never requested for exposures of type "zero" or "dark". The parameter **filtype** should always read "instrument".

The remaining parameters will be used in the future for improved telescope/instrument focus interaction. For now, keep **setfocus** = "no".

3.3 wheel1/2/3

*These parameter sets are **not** used with the MOS Bench Spectrograph.*

These parameter sets are used with the Imager FSA. The FSA has three (3) different filter wheels but only one (1) filter wheel can be inserted and used at a time. The FSA software can sense which wheel is actually in use and will warn the observer if the observer specifies the wrong filter wheel.

The **wheel1/2/3** psets allow the observer to specify the name and focus offset for each filter position in each wheel. This allows filters to be referred to by name instead of having to remember which filter is in which slot, in which wheel. They also contain the corrections to the telescope focus which must be made when each filter is used. In the future, these values will be used to automatically adjust the telescope focus when filters are changed. For now, these values are merely stored in the pset file for later reference.

When editing **instrpars** you can access these parameter sets by using the arrow keys to move down to the correct line and then type **:e**. Alternatively you can edit these parameter files by typing e.g.

`cl> epar wheel1` -OR-

```
cl> wheel1
```

Make certain that no extraneous characters appear in the value field of **wheel1**, **wheel2** or **wheel3** after exiting; if they do just enter a null string ("") as the value. Whichever method you choose you will see something like this,

```
       I R A F

     Image Reduction and Analysis Facility

 PACKAGE = fsa
 TASK = wheel1

 (id1 = u) Short identifier for filter in position 1
 (name1 = Harris_U) Full name of filter in position 1
 (focus1 = 40) Focus value for filter 1

 (id2 = B) Short identifier for filter in position 2
 (name2 = Harris_B) Full name of filter in position 2
 (focus2 = 10) Focus value for filter 2

 (id3 = V) Short identifier for filter in position 3
 (name3 = Harris_V) Full name of filter in position 3
 (focus3 = 0) Focus value for filter 3

 ....

 (id8 = drk) Short identifier for filter in position 8
 (name8 = Dark) Full name of filter in position 8
 (focus8 = 0) Focus value for filter 8
```

Each group of three parameters corresponds to one slot in the filter wheel. The parameter **id#** is the short identification for the filter that you should type when prompted for a filter position by **observe**. It should be short, should be unique amongst the filters in the wheel and must not contain any white space or other weird characters. One exception to the uniqueness requirement is that any empty positions may be given the same name e.g. "clear" (but *not* a null "" or blank " " name) provided you don't care which one is actually used. The value of **name#** is what will be recorded in the image header. It can not contain the special character "~" and blanks will appear as underscores in the header.

The **focus#** correction feature is not implemented in the current version of the software. The corresponding filter focus offset parameters should be set to zero for now.

3.3.1 Setting Up the Imager FSA Without Taking an Exposure - *Instrument*

If you want to say, move the filter wheel, without taking a CCD exposure, use **instrument**. This task will prompt you for positions for each motor and move it to that position, just as observe does. However, **instrument** obeys the **setfilters** and **setfocus** parameters in **obspars** and hence will only move those motors you have enabled. More often than not it is precisely those motors you have disabled in **obspars** that you want to move. For instance you may normally have **filter2** disabled (**setfilters=one)**, but want to switch to a filter on a different wheel and switch back at the end. If you want to override **setfilters** for a single invocation of **instrument** proceed as follows:

```
cl> instrument setfilters=two

Filter in wheel two (u): cb
```

The **setfocus** parameter can be overridden on the command line in the same way.

4.0 FSA Package

The FSA (Filter/Shutter Assembly) Arcon/Iraf package allows the user to enter commands through the IRAF interface to manipulate the FSA. *Most observers will never have to use these commands.* The following commands are available:

- **initfilter** - will reinitialize the filter wheel
- **setfilter** - set filter wheel to position N
- **initshutter** - will reinitialize the shutter
- **setshutter** - will set the shutter to a particular mode - either normal, xfer or fast (these depend on the exposure time)
- **expshutter** - this sends the exposure time to the FSA and changes it's mode
- **rebootfilter** - will reboot the filter wheel
- **rebootshutter** - will reboot the shutter

The **observe** command will automatically prompt the user for the filter and will send the exposure time and the chosen filter to the FSA.

Note: if the MPGrouter is down, these commands will not work through the Iraf Acquisition Window. They will either timeout or fail.

5.0. Focusing the Imager

5.1. Establishing the Initial Imager Focus

There are two ways to establish the current Imager focus: **wavefront curvature sensing** and a **manual focus exposure**. The former method is recommended for the start of the night or when changing to a filter with an unknown filter offset. The latter method is quicker during the night when the observer is trying to fine-tune the focus in a sequence of exposures using the same filter or when changing to a filter where the filter offset is known.

Wavefront curvature sensing is easy but time consuming - it takes about 15 - 30 minutes. This is often time well-spent if you've become really confused (often the case in these early days). Here's the easy part: just ask the duty telescope operator to do this. Remember to tell them what filter to use!

Manual focus exposures are a little more involved but often faster. Some coordinating with the telescope operator is still needed. The basic idea is to take obtain multiple images over a range of focus on the same CCD frame. *Note that 10 microns is about the right minimal secondary movement amount.* Here's a simple procedure:

1. Ask the operator to adjust to secondary position about -30 microns.
2. Start a 105 sec exposure in the desired filter.
3. After 15 sec, stop the exposure with the *pause* command.
4. Ask the operator to move the telescope 20 arcsecs North and the focus +10 microns.
5. Re-start the exposure with the *resume* command.
6. After 15 sec, stop the exposure with the *pause* command.
7. Ask the operatore to move the telescope 10 arcsecs North and the focus +10 microns.
8. Re-start the exposure with the *resume* command.
9. Repeat steps 6, 7, and 8 until the time is up.
10. After the exposure is read-out, use *imexamine* to determine which images were the roundest and sharpest. Ask the operators for assistance with *imexamine* if necessary.

This manual focus exposure procedure will be automated in the next major Arcon software upgrade.

5.2. Maintaining the Imager Focus

Because WIYN is a very lightweight telescope, it mechanically responds quite rapidly to ambient temperature changes: a 1 deg C change corresponds to about a 40 micron expansion or contraction. The telescope often cools by 2-7 degrees C rapidly after sunset and then about 1 deg per hour for the rest of the night. This corresponds to focus changes of 80 - 320 microns after sunset and 40 microns per hour after that. Since a 10 micron change significantly defocuses the telescope, compensating for these large changes is very important in maintaining focus.

To deal with this situation, the telescope trusses have been wired with thermocouples which report the truss temperatures once every 10 secs. We have determined empirically that:

- negative 1 deg C ambient temperature change (cooler) => +40 micron focus change
- positive 1 deg C ambient temperature change (warmer) => -40 micron focus change

This calibration is preliminary and will be refined as soon as possible.

The recommended procedure is to monitor the telescope temperature and adjust the telescope focus by the appropriate amount when a significant temperature change has occured. Please ask the operator to show you how to monitor the telescope temperature.

One of the WIYN Project's highest engineering priorities is to automate this process so that it is transparent to the user.

5.3. Filter Focus Offsets

To first order, the focus offsets between the broad-band filters typically used at WIYN are determined by the filter thicknesses. Thus, these focus offsets are fixed with time.

We have not determined reliable filter focus offsets at this writing. For now, observers should pick a filter to be their baseline filter and then determine offsets to their other filters. Such offsets should be reported to Dave Sawyer (dsawyer@noao.edu) and Dave Silva (silva@noao.edu) so we can begin to compile an offset database.

6.0 Router Related Issues

The **MPGrouter** is the source of all information from other WIYN specific instruments. Special clients are written for each instrument. These clients send out status and instrument specific information and subscribe to other instrument information that has been made available. The maximum frequency of updates to this information is on a per second basis. The information should be within a 2 second range of accuracy, unless messages have been dropped. Dropped messages are generally due to a CPU being too busy to read them.

The **MPG MSG** icon contains the most recent detected status of the MPGrouter. It is only noticed during the picture taking phase, where the process of getting and reformatting header information is done. The process is called **icsInfo**. It subscribes to **FSA** information as well as the **TCS** data streams.

The warnings that it issues are:

1. If the TCS information is not being updated between pictures you will get a message indicating that you need to have the operator check to verify that messages are being sent to your client.
2. If the MPGrouter comes down, and you have been able to reconnect successfully, then you will see a message saying that you can continue to take data.
3. If the MPGrouter comes down, but you are unable to reconnect, then you will see a message that you will need to restart Arcon when the MPGrouter has recovered.
4. If the MPGrouter is not running when you bring-up Arcon, then you will see an alert that tells you to Restart Arcon when the MPGrouter is running.

You can iconify this message by clicking on the **CONTINUE** button.

You can collect data if the MPGrouter is not running, but you will not get TCS, FSA, Hydra, or Bench Spectrograph headers. Furthermore, FSA commands entered through the IRAF interface will not be executed. This causes long delays, waiting for commands to timeout.

Appendix A: Arcon Software Task Cribsheet Summary

Data taking commands

- **observe** - Take one or more exposures prompting for type
- **dark** - Take one or more dark exposures
- **comp** - Take one or more comparison exposures *(Bench CCD only)*
- **dflat** - Take one or more dome flat exposures
- **object** - Take one or more object exposures
- **sflat** - Take one or more sky flat exposures
- **zero** - Take one or more zero exposures
- **focus** - Take a focus frame
- **more** - Take more exposures of the previous type
- **doobs** - Take multiple exposures in multiple filters

Exposure Control Commands

- **abort** - Stop exposure and do not readout detector
- **stop** - Stop exposure and readout detector
- **tchange** - Change exposure time
- **pause** - Pause current exposure
- **resume** - resume paused exposure

Parameter Sets

- **obspars** - Observing parameters
- **detpar** - Detector parameters
- **instrpars** - Instrument parameters *(Rarely used.)*
- **wheel1** - filter info parameters for Imager FSA wheel1
- **wheel2** - filter info parameters for Imager FSA wheel2
- **wheel3** - filter info parameters for Imager FSA wheel3

FSA Setup Control

- **instrument** - Generic instrument control based on instrpars pset

Loaded packages

- **arcon** - Automatically loaded main Arcon package
- **FSA** - Automatically loaded FSA Filter Shutter Assembly package (Imager only).
- **connect** - Task to make connection to detector controller (automatically done by FSAetc)
- **disconnect** - Task to break connection to detector controller (automatically done on bye from FSA)

Appendix B: Logging in and Logging Out

B.1. Computers and Peripherals

Figure 1 shows a picture of the WIYN control room.

Figure 1 shows the control room at **WIYN**. Sam Barden, the Hydra Project Scientist, is sitting in front of the monitor for **oatmeal**, the Hydra Sun workstation. The Imager Sun workstation **pearl** is in the far corner near the empty blue chair to Sam's right. Sam is using **Oatmeal** and next to him on his left is **Vanilla** another data acquisition computer used with the Bench Spectograph and Hydra.

B.2. A Guided Tour of the Arcon Windows

The Imager Harcon system is run from the **pearl** workstation while the Bench Spectrograph CCD is run from the **vanilla** workstation. The current observer logins and passwords are posted on the machines.

When the observer logs in, an OpenWindows environment is loaded and the Arcon-IRAF interface software is executed automatically.

Several windows will come up including a IRAF Data Acquisition and a IRAF Reduction window. The controller software will automatically be down-loaded as well as any appropriate CCD controller parameters specific to the installed chip. Also, the **FSA** IRAF package will be loaded into the IRAF Acquisition window CL.

It will take several minutes for the hardware and software to be ready. When it finishes the message "Connecting to the controller..." will appear, followed by "Data reader restarting ...." and several messages associated with the downloading of the detpars waveform. *Wait until you see the "**cl>** prompt" in the data acquisition screen before continuing.*

```
      abort       focus         pictitle      setshutter
      ccdinfo     initfilter    postarch      sflat
      comp        initshutter   postkill      shutter
      connect     instrpars@    preview       stop
      dark        instrument    rebootfilter  tchange
      detcomp     more          rebootshutter teloffset
      detparm@    movie         recover       telpars@
      detpars     object        resume        wheel1@
      dflat       observe       rtdpars@      wheel2@
      disconnect  obspars@      setdetector   wheel3@
      doobs       pause         setfilter     zero
      expshutter  pflat         setrtd

cl>
```

Loading the instrument package automatically establishes the connection to the controller, as indicated by the "Connecting to the Controller...." message hidden amongst the package menus. Exiting from the package with **bye** or (usually inadvertently!) **CTRL-Z** breaks the connection.

NOTE: You can not take data if the connection is broken.

Once the initialization procedure has been completed, your screen should appear somewhat like Figure 2.

Figure 2 shows a picture of the Arcon Data Acquisition Environment

The key windows for taking data with **Arcon**, identified by the names given in their title bars, are as follows:

Arcon CONSOLE - As its name implies this window serves as a console for **Arcon**. While you are observing you will see many messages appear here, only some of which will be repeated in the IRAF acquisition window. This window is hidden behind the Data Acquisition screen and if it is distracting, you can *close* it. But don't *quit* from it. In the event that something goes wrong, the diagnostic messages appearing in this window may tell you (or at least us) what happened.

Arcon STATUS - This very important window gives several lines of information summarizing the status of the controller, the instrument, and any ongoing exposures (see Figure 2). The first line shows what the controller is currently doing. if you have just brought the system up, it should read "CONTINUOUSLY_ERASING" indicating that the CCD is idle and is continuously running the erase cycle; if it doesn't, chances are something went wrong in the initialization and you should seek help. During exposures this line should read "INTEGRATING", and should change to "READING" as the CCD is read out. Other messages which may occur will be described later as appropriate. Just below the status line are counters showing the number of seconds left in the current exposure, the number of exposures left in the current sequence, and, during read out, the number of buffers of data successfully transferred to the Sun. Also shown are parameters of the current exposure such as the title, picture name etc. Finally, for the Harcon the bottom line contains some important telemetry information. This includes the temperatures and voltages of the Dewar and CCD as well as the CHIP ID number.

COUNTDOWN -This is the very small window with the very large font. It provides a copy of the exposure time counter for the visually handicapped.

IRAF Acquisition - This black colored window is one of two running the IRAF cl. We recommend you type all data taking commands here, so that your data taking and data reduction activities are well separated and don't interfere with one another.

IRAF Reductions - This window (colored an off-white) is the other IRAF window. We suggest that normal IRAF commands, used to examine or reduce your data are entered in this window. The parameters used in this IRAF environment are NOT the same as those used when taking data in the Data Acquisition window.

IRAF Display Window - "Ximtool" window will be used when displaying your images from IRAF. This resides on the 2nd monitor.

The following 3 windows display messages associated with **MPGrouter** messages.

- **icsInfo** - this client gathers and converts information from other instruments into IRAF format. It will also create the **MPG MSG** icon which will report changes in the router when detected. It's window is behind the Data Reduction window.
- **Arcon Client (arClient)** - this client connects **IRAF Acquisition Interface** to the **MPGrouter**. It gets information as well as sends it. All **FSA** tasks from the FSA Arcon/IRAF package send a command to this process.
- **FSA Motor** - this client receives commands from the arClient and executes them. The commands are sent to the Filter Shutter Assembly over a serial line. The state of the FSA is also made available through the router. All the commands needed to manipulate the FSA are available in the IRAF Data Acquisition window. See section 4.0 for more information concerning the FSA.

A further word about image directories is in order at this point. Your data acquisition machines has roughly 9 Gb bulk data storage. On each of these there will be a directory, /data1, where this space is mounted. At the start of your run all these disks will be cleared. However, you may have no trouble filling the available space in a few nights of typical observing. From time to time you should use the command **disks** to see how much space is free on each disk. **Arcon** uses the value of the IRAF environment variable **imdir** in order to decide where to write the pixel files for newly created images.

B.3. Shutting Down and Logging Out

Before you log out, for whatever reason, you must first stop the Arcon related processes. To do this, bring up the background menu and follow the Arcon submenu. Release the mouse button on **Arcon Exit**. The processes that will be removed are all transputer, arcon/iraf and router executables and their corresponding windows. " IRAF

Acquisition" is setup to automatically disconnect upon exit or logout. This allows for a clean download of transputer code when the Arcon is used again.

Appendix C: What to do if things go wrong

C.1. SUN in a strange State

If you have a suspicion that the system is somewhat flakey you can verify that everything is cleaned up after exiting Arcon by typing:

```
pearl% NexUp
No match.
No match.
No match.
```

If you get "No match." three times in a row, as above, all is well. If instead you see something like

```
pearl% NexUp

root      2227  0.0 3.0 2664  844 p6 S    17:32   0:05 muxnex
arcon     2229  0.0  .6  172  160 p2 S    17:32   0:00 arsh c0e2
arcon     2228  0.0  .0  144    0 p2 IW   17:32   0:00 arsh c0e2

/dev/nexc0

Nomatch.

rwrw    1 arcon           0 Jul 13 17:32 /tmp/xpim2229.1
rwrwrw  1 arcon           0 Jul 13 17:33 /tmp/xpim2245.1
```

then there are some leftover processes which you must kill by hand as follows,

```
pear1% kill -9 2227 2229 2228
```

In this command the list of numbers after the -9 are the pid's of the leftover processes as displayed by **NexUp**. The files with names like /tmp/xpim2229.1 are the spool files used by **Arcon** to transfer data to the Sun. If any of these are owned by you and have a size other than zero (the number just before that date), then they may contain your missing data! See Appendix C section 3 for information on how to retrieve this. It is common to see a few zero length files as shown in the example, but these can be ignored.

Now you can exit from the windowing system and log out completely, by moving the mouse to a blank area of the screen, then hold down the right mouse button, and select "**exit**" from the menu which will appear.

C.2. Warm Starts

Rather more often than we would like, something or other happens which causes the system to hang requiring that the software be reloaded.

Before doing so it is worth testing to see if the problem is confined to the IRAF interface layer, by proceeding as follows:

```
cl> flpr
```

This flushes out any brain-damaged executables locked into the IRAF process cache. Having done this, test to see if the problem has gone away by taking a "zero" frame.

If this fails, or if the status window has stopped functioning, then the problem is probably in **Arcon** itself and it is best to reload the software from scratch. This takes very little time. There are 2 options for restarting Arcon.

1. If the CPU is not maxed out then:

   ```
   Bring up the background menu, follow the Arcon submenu and release the
   mouse on Arcon ReStart.
   ```

2. If the CPU is pegged then:

   ```
   Bring up the background menu, follow the Arcon submenu, and follow the
   CPU Pegged Options:

   Release the mouse on Iraf/Arcon Kill. After all processes disappear,
   follow the CPU Pegged Options menu again and release on Iraf/Arcon Start.
   This will start new Arcon and Iraf processes.

   If the CPU is still pegged, then follow the instructions for logging out
   and then log back in.
   ```

C.2.2. If All Else Fails

Just once in a while a problem will occur which just refuses to go away even when you reload the **Arcon** software. This may be due to a hardware failure. However, it may also be that the Sun has gotten irremediably confused, in which case the rather drastic step of rebooting it may be needed. If this is necessary, contact observer support. They can show you how to reboot the Sun safely, cleanly, and without having to know the superuser password.

C.3. What to do if Your Picture Doesn't Show Up: *Recover*

Sometimes **Arcon** will successfully transfer your data to the spool file on /tmp but the picread program which converts this to an IRAF image will fail. Usually there is an error message, but you should be suspicious that this has happened if the exposure completes normally, but you can't find the output image. The command **recover** will help you to retrieve your valuable images in such cases. It searches /tmp for any spool files owned by you and for any of these that are complete will show the FITS header and ask if you want to recover the image or not. If you reply yes then picread will be run to convert the xpim file to an IRAF image. A single spool file occasionally contains more than one image. In this case **recover** will list how many images are present, but will only show you the header of the first. All images will be recovered if you tell the program to go ahead. Spool files are not removed from /tmp until successfully converted by picread. However, we suggest you run **recover** immediately if you encounter this problem, so that you don't forget later, and so that /tmp doesn't get filled with unprocessed raw data files.

Sometimes a failure occurs during the transfer of the data from the controller to the Sun. In this case a partial spool file results, and there is no way to resurrect the data. **Recover** will give you the option of deleting any such files, and you should do so to avoid filling /tmp with debris.

Appendix D: Changing Image Directories

You can change where your image data is stored "on the fly" with the command

```
cl> reset imdir = /data1/whatever/
```

Note that the trailing "/" character is necessary. This variable will be restored to its original default value when you log out of IRAF. Also the values in the acquisition and reduction windows are independent and must be set separately. To change **imdir** permanently and insure the two windows use the same value, you must edit your .login file as follows,

```
cl> edit home$.login
```

```
# PEARL IMAGE .LOGIN DEFAULT FILE

..........

# Image pixel directory definition. This is the ONLY place where this

# variable should be defined and changed.

setenv imdir "/data1/whatever/" <----- Change this line as required
```

You must then log out of Arcon and log back in, for the change to take effect.

It is common practice at KPNO to set the IRAF image (pixel) directory to "HDR$pixels/", i.e.

```
cl> reset imdir = /data1/whatever/
```

This option assures that your pixels files will appear in a sub-directory named *pixels* below the directory where the IRAF header files are created. *We urge the observer to adopt this directory structure style.*

Appendix E: Postprocess

There is an option to automatically perform a **IRAF** process on the newly created image. This is fondly known as **postproc**. To choose this option, set **obspars.postpic** (see Section 3.2) to `postproc $image` . If you have chosen this option via the obspars parameters, at the end of the picture creation, you will see the message "[Running post processing command: ....]" and the picture will automatically be displayed in the ximtool window.

There is another option to run a process after the entire sequence. It works in a similar fashion and is activated by setting the **obspars.postseq** parameter to the script you want executed.

Note: if you want a special task to be executed, it must be defined as an IRAF task in the usual way. See the appropriate IRAF documentation for further assistance. You will then have to set the **obspars.postpic** or **obspars. postseq** parameters as desired.

```
Examples of setting the obspars parameters:

(postpic = "display $image 1"   ) Post processing command for single exposures
(postseq = "imcombine $sequence") Post processing command for sequences

In the above example, for a single exposure both the postpic and postseq
scripts will be run.  The postseq script is always run.

The possible substitution strings are:

$image        -replaced by name of last image readout
$sequence     -replaced by list of images in sequence
$picture      -replacec by picture number of last image readout
$npics        -replaced by number of pictures in sequence
$type         -replaced by exposure type
$title        -replaced by exposure title

examples:

display $image 1                         - display last image
imcombine $sequence                      - combine all images in sequence
hedit $image i_title \"$title - $type\"  - change image title to include type
                                           (quotes must be \ escaped)
postproc $image                          - the nominal default KPNO
                                           postproc script
```

Warning: *no check is made of the command syntax or it's validity, or that the necessary packages are loaded until it is executed by the cl. Errors in the execution may produce incomprehensible error messages from IRAF. To clear this out type* ***flpr***

An information sideline, from within IRAF, if you cd to the directory containing your headers you can do the following types of commands:

```
imexamine @lastpic     - display and examine the last exposure
imcombine @lastseq     - combine last sequence
```

These are special functions that accept @file input. The lastpic has the last exposure name in it. Similarly, the lastseq has a list of the last sequence taken in it.

The **restart** parameter in obspars will determine how the postprocess is run. If it is set to "yes", then a new postprocessing job is created for each new image. This can create too many IRAF background processes. If too many jobs are active, the task **postkill** can be used to clear them out. If you do run out of batch slots, you will be warned that it was not possible to execute the postprocess command.

If **restart** is set to "no" then you need to remain in the directory that you started in, when using postprocessing. Also, the postprocessing script can not be changed unless you restart Arcon after the change is made. This option loads in the script initially and uses the files **lastseq** and **lastpic** in the current directory to perform the desired script(s).

Appendix F: Writing Your Data To Tape

F.1. Making a FITS tape using *wfits*

So now that you have some data, you probably want to save it to tape using the IRAF command **wfits**.

First, you must allocate a tape drive. At WIYN, there are both Exabyte and DAT tape drives. The Exabytes are known as **mtb** and the DAT drives are known as **mtc**. (You can find these names printed on each drive unit; you can also run the IRAF command **devices** to list both the IRAF and Unix names of all the I/O devices on the mountain.) Do an

```
cl> allocate mtb
```

to allocate the Exabyte tape drive at WIYN (use mtc for the DAT drives).

Since all of the KPNO IRAF data acquisition systems subscribe to IRAF networking, you can in fact use any tape drive in any dome, although you need to check with the appropriate astronomer if you want to use someone else's tape drive---be nice!. To allocate a drive elsewhere you might do something like a

```
cl> allocate lapis!mtb
```

The parameters for **wfits** are:

```
   iraf_files = "@savethisstuff" IRAF images
   fits_files = "mtb"           FITS filename
      newtape = yes             Blank tape?
       bscale = 1.              FITS bscale
        bzero = 0.              FITS bzero
  (make_image = yes)            Create a FITS image?
 (long_header = no)             Print FITS header cards?
(short_header = yes)            Print short header?
      (bitpix = 0 )             FITS bits per pixel
(blocking_fac = 0)              FITS tape blocking factor
       (scale = yes)            Scale data?
   (autoscale = yes)            Auto_scaling?
        (mode = "ql")
```

The crucial parameters are:

- **iraf files** - This refers to the image name(s) that you wish to save, e.g., ``a0002,a0005'' if you wish to only save those two files. Instead, you can also substitute an "at" file here (such as shown in the example). An "at" file is simply a text file that contains the names (one per line) of each image. You can easily create such a file by **files *.imh > savethisstuff**.
- **fits files** - This refers to the tape drive you wish to use. In the case of the Exabytes or DATS this can simply be the same as the device name that you used in **allocate**.
- **newtape** - This is, as you may well imagine, a critical parameter to get right. If you want to start at the beginning of the tape, writing over any data that might be present, then answer **yes** to this; otherwise it will search for the end of the data on the tape before writing.
- **bitpix, scale, and autoscale** - By leaving these values at their default values (**bitpix=0, scale=yes, autoscale=yes**) the data will be written to tape in such a way that you are never losing any precision. Nothing will be scaled, and raw CCD data will be written as unscaled twos-complement 32-bit integers, while floating point data will be written as 32-bit, unscaled, IEEE-standard reals. Change these at your own risk!
- **blocking fac** - The default, 0, results in ``maximum'' blocking (10) for both Exabyte and 9-track tapes. If you are planning to read the data with a non-IRAF FITS reader, you may want to reset this to 1; otherwise, leave it at 0 for maximum tape writing efficiency.

In order to check to see what is on the tape, you can list the titles quite easily. Simply do a

```
cl> rfits mtb 1-999 make- short+ old+
```

to see what's there. To direct this output into a file, you can add a tapelist to the end, and then you can print that list on the printer by a simple **lprint tapelist**. Alternatively, you could choose to read a single file back onto disk to examine:

```
cl> rfits mtb 15 junk make+ old+
```

will create an image with the original name. (Note that even if you specify **old+** you must still give the name of a legal temporary file in the output file position, i.e. junk in this example.) If the same image name already exists on disk, you should instead use **old-** to create an image named junk , or whatever you wish.

As you write your tape, you will get a file-by-file account of what is going on the tape. When you come to the end of the tape, you will find that there is a message saying that the file was NOT completely written to tape. An easy way to continue at this point is to edit your "at" file to remove the names of the files that were successfully written to tape. Rewind the tape with **rew mtb** , mount the new tape, and then re-run **wfits** , remembering to set **newtape=yes**.

NOTE: If you do write additional files to an ``old tape'' (one containing useful data but which had previously been removed from the drive), make certain that the software (IRAF and Unix) is aware that the tape has been rewound before starting to write to the tape----or your old data may be overwritten! To safegaurd against this possibility we suggest that you ALWAYS swap tapes by first:

```
cl> deallocate mtb (or mtc or ...)
```

Physically swap tapes

```
cl> allocate mtb (or mtc or ...)
```

F.2. Safe Taping

We remind the observer that s/he, and s/he alone, is responsible for keeping one's bits safe. We recommend the following before deleting any data from disk:

1. Each night write data to whatever media you like using **wfits**
2. Read the tape using **rfits mta make- old+** to substantiate everything is there.
3. Deallocate the drive, remove the tape, and *stick it under your pillow*.
4. Make a second copy of your tape. (This tape could be an accumulative copy of the data throughout your run.) Check this tape with **rfits**!
5. Only now delete the data from disk if necessary.

F.3. Save-the-Bits!

Transparent to the observer, all CCD images are automatically "archived" onto Exabyte tapes. This program should not encourage lax taping procedures, and we strongly emphasize the need for the "safe taping" procedures above. But if you ever do need to recover a night's worth of data, take heart! You can contact Ed Carder if you need this service (ecarder@noao.edu).

Appendix G: Summary of Useful IRAF Commands for Quick Look

This section has no pretensions of being a tutorial to IRAF, but it is our intention to call your attention to or remind you of some of the more useful IRAF commands for examining your data. Use the IRAF help command and/or poke around to become familiar with these commands.

- **diskspace** - check remaining space on disk
- **dir** - lists the files in current directory

- **imcopy** - copies an image, or subraster of an image, to another image
- **imrename** - renames an image
- **imdelete** - delete images

- **imheader** - list image headers
- **hedit** - edit the header parameters of an image
- **hselect** - list specific items from the image headers

- **display** - displays an image on the ximtool window
- **imexamine** - general image analysis utility (row and column plots, radial profiles, surface plots, contour plots, image statistics)

- **imhist** - histogram the pixel values in an image
- **imstat** - detailed statistics on an image
- **implot** - plot rows, columns, and averages of these
- **splot** - plot a row from an image with spectral analysis tools

- **imarith** - arithmetic operations between images ( )
- **imcombine** - combines images (such as averaging or median)

- **allocate** - allocates a tape drive
- **deallocate** - deallocates a tape drive (necessary in order to logout)
- **devstatus** - reveals what IRAF thinks about the tape
- **devices** - lists all the I/O devices and Unix names
- **wfits** - write a FITS format image to tape or disk
- **rfits** - reads a FITS format image from tape or disk
- **mtexamine** - examines a magnetic tape and divulges interesting facts
- **files** - generate a list of files (useful with wfits)

Appendix H: Using *ximtool*

The IRAF image display window is ``ximtool'', and combines the best features of Sunview/IMTOOL with SAOIMAGE. To load an image into the display, simply type **display image 1** , where the ``1'' denotes display frame number 1. (Up to 4 images can be loaded in frames 1-4.) This command uses many defaults that sometimes need to be changed, depending on the data and your needs.

Since the largest image that physically fits on the screen contains only about 1000 pixels on a side, and since the WIYN CCDs are considerably bigger than this, there is some trickiness to how you might want to display your data. Basically you have two options:

- Set the buffer size (*set stdimage=imtname*) to the full size of the image, and display to it using *display imagename 1 fill-*. The **fill-** will assure that each pixel in the image gets mapped into a single pixel in the display. For our large chips, however, only a subsection of the image will fit on the displayed area. Fortunately there is a panner box in the upper right hand corner of the window which will show you the entire image with a bright green box indicating which subregion is displayed on the main part of the window. The advantage of this is that when you see a pixel, it is really a pixel, and you can easily evaluate things like bad pixels and columns. The disadvantage is that you cannot see the entire image at once, but must move the panner box in order to change the region being shown.
- Set the buffer size (*set stdimage=imtname*) to something small enough to fit on the screen but display to it using *display imagename 1 fill+*. The **fill+** means that every pixel in the image will contribute to the intensity of each pixel in the display, but not 1:1. In other words, if you were displaying a 1024 x 1024 image into a frame buffer that was 512 x 512 large, using fill+ would be the same as averaging ajacent pixels. The advantage of this is that you will see the entire frame without having to move the panner box; the disadvantage will be that even if you ``zoom'', you will not be seeing actual individual pixels.

Phil Massey recommends option (1) for spectroscopic applications but (2) for direct imaging. One of the other authors prefers to do both, using (2) [see the full frame] in the ``Data Acquisition'' window, so that images will be automatically displayed showing the full field, but setting the defaults in the ``Data Reduction'' window using (1) so that each pixel can be cherished by redisplaying an image from there.

Once the image is loaded, you can adjust the brightness and contrast parameters by moving the mouse in the window while holding down the right-hand button. Negative contrast (black stars on white background) is realized by moving the mouse to the upper half of the ximtool window; positive contrast is obtained by moving the mouse to the lower half of the window. Higher contrast is achieved by moving the mouse vertically away from the center. The image is made darker (blacker) by moving the mouse to the upper left (negative contrast) or lower right (positive contrast). The most extreme displays are usually obtained when the mouse is in one of these two corners.

You can learn the approximate pixel location of the mouse by reading the value in the box at lower right. The intensity is also displayed there. If the intensity read-out has a "+" or "-" next to it, the pixel is either brighter or fainter than the display command has scaled the data. To change the scaling from the default, use the display command as follows:

```
cl> display image 1 zscale- zrange- z1=0 z2=2000
```

to set the scaling is now set to 0-2000 rather than the automatic scaling printed out previously when display was executed.

You may also zoom the picture using the middle mouse button. Press the button once and the image will move in an attempt to center up on the mouse cursor. Press it again without moving the cursor and the image will zoom up a factor of 2. Pressing the middle button 3 more times will zoom the picture up to a factor of 4, 8, and back to 1 (normal). If you move the cursor while zoomed and then press the middle button, the picture will pan to center up on the cursor.

You can also pan by simply grabbing the green outline in the panner box using the left mouse button, and moving it to outline the part of the image you would like.

The controls on the upper right allow one to flip the image left and right, up or down, change frames (the frame number is displayed between the two fat arrows), and, best of all, open a control pannel (left most icon). The control pannel allows you to turn the coordinates box and panners on and off, flip, blink, and much, much more.

If you find that you have somehow mysteriously added little green boxes on your image, you can get rid of these by placing the cursor in such an extraneous box, holding down the right-most mouse button, and selecting ``Destroy''.

Appendix G: Using *imexamine*

The **imexamine** task provides some of the most powerful diagnostic quick-look tools within IRAF. If an image that you wish to examine is already displayed in the ximtool window, simply type **imexamine**. Wait patiently *without moving the mouse* and a blinking, round cursor will appear on the display. (If you lose patience, try pressing the ``L5'' key without moving the cursor.) Place the cursor over a star, and strike the **r** key, and you will be presented with a radial plot of the star, along with the values of a fit to the stellar profile. The last number displayed is the FWHM in pixels, quite useful for determining the best focus. Other very useful commands include **l** for making a line plot at the position of the cursor, and **c** for making a column plot at the position of the cursor. Other useful cursor strokes are shown below.

- **r** - Make a radial profile of the star near the cursor. The FWHM will be the last number shown in the plot.
- **a** - Print FWHM without showing the radial plot.
- **c** - Plot the column nearest the image cursor
- **l** - Plot the line nearest the image cursor
- **j** - Fit a 1-d Gaussian in the x direction, centered near the cursor
- **k** - Fit a 1-d Gaussian in the y direction, centered near the cursor
- **m** - Print the statistics in a box around the image cursor
- **e** - Make a contour plot of a region around the image cursor
- **h** - Plot the histogram of a region around the image cursor
- **s** - Make a surface plot of a region around the image cursor
- **:c N** - Plot column N when in graphics mode
- **:l N** - Plot line N when in graphics mode
- **:naverage M** - ave M columns (or lines) during plots
- **x** - Print the x, y, z values of the pixel nearest the image cursor
- **z** - Print a 10 by 10 grid of pixels around the image cursor
- **o** - Overplot
- **g** - Go to the graphics window from the ximtool window
- **i** - Return to the ximtool window from the graphics window
- **?** - Print help
- **q** - Quit **imexamine**
- **:epar r** - Edit the radial profile plot parameters
- **:epar c** - Edit the column plot parameters
- **:epar e** - Edit the contour plot parameters
- **:epar h** - Edit the histogram plot parameters
- **:epar l** - Edit the line plot parameters
- **:epar s** - Edit the surface plot parameters

Appendix I: IRAF Miscellanea

I.1. Analyzing Direct Focus Frames with *kpnofocus*

Analyzing direct imaging focus frames has now been made fun and easy thanks to the new *kpnofocus* routine. Simply run *kpnofocus*, giving as input the name of the focus frame. Next, mark the top image in a multiple focus exposure using the ``g'' key. You will have to wait a few seconds, but you will then be presented with a plot of FWHM vs. focus value. Do this for a few stars and then exit using ``q''. This routine must be loaded from the **nmisc** package.

I.2. Focusing a Spectrograph with *specfocus*

Frank Valdes has provided a very useful routine to aid us in determining the best focus of a spectrograph using a series of exposures of comparison lines. The assumption is that the collimator or camera focus has been changed between each of the exposures. The routine works by cross-correlation, and provides an easy measure of the full-width-at-half-maximum (fwhm) of the images. In addition, one can specify that the dispersion axis be broken into multiple samples to determine if the best focus is wavelength dependent, and/or the spatial axis can be broken into multiple samples to determine if the best focus varies across the spatial axis.

The simplest session of this program looks like this:

```
nl> lpar specfocus
       images = "d*.imh"        List of images
       (focus = "1430x10")      Focus values
    (corwidth = 20)             Correlation width
       (level = 0.5)            Percent or fraction of peak for width measureme
      (shifts = yes)            Compute shifts across the dispersion?\n
    (dispaxis = 2)              Dispersion axis (long slit only)
    (nspectra = 1)              Number of spectral samples (long slit only)
       (ndisp = 1)              Number of dispersion samples
       (slit1 = 5)              Lower slit edge
       (slit2 = 50)             Upper slit edge\n
     (logfile = "logfile")      Logfile
        (mode = "ql")

SPECFOCUS: NOAO/IRAF V2.10EXPORT feed@indigo Sat 19:37:35 19-Sep-92
  Best average focus at 1465. with average width of 1.41 at 50% of peak

  -- Average Over All Samples

                                 Image  Focus  Width
                             d0001.imh  1430.   2.86
                             d0002.imh  1440.   2.23
                             d0003.imh  1450.   1.66
                             d0004.imh  1460.   1.44
                             d0005.imh  1470.   1.44
                             d0006.imh  1480.   1.89
                             d0007.imh  1490.   2.54

  -- Image d0004.imh at Focus 1460. --
```

The dispersion axis goes along columns (**dispaxis=2**), and the slit covers columns 5 through 50. In addition, there is a very sophisticated graphical display that helps in interpreting the results in the case of multiple sampling along the dispersion/spatial axis; read the **help** page for **specfocus**. Currently specfocus may be loaded from the **nlocal** package.

I.3. Determining the gain and readnoise using *findgain*

At CTIO it is common to measure the CCD gain and read-noise whenever a CCD is installed on the telescope. Rob Seaman has provided a useful script called **findgain** that will provide good estimates of the gain and read-noise using two flat-field exposures and two bias frames. The frames should not have been processed or combined in any manner. The astronomer should specify a section of the image over which the flat-field is relatively constant. Additional details can be found by reading the help page for **findgain** . The routine currently lives in the **nproto** package.

I.4. Determining the shutter correction time with *findshutcorr*

The actual exposure time of your image may differ by some fraction of a second from the requested exposure time due to the fact that no shutter can open or close instantaneously.

If you are attempting absolute photometry or spectrophotometry, and plan short enough exposure times that you are worried about the shutter correction, there is a simple routine in IRAF called findshutcor which will help you evaluate the correction. Written by Rob Seaman, the routine determines the shutter correction from a series of flat-field exposures of varying lengths (1 sec - 20 sec, say). You must be careful not to have saturated on the longest exposures, of course, and the frames must have been processed for removing the overscan level, but not flat-fielded. In taking the data one must also carefully take into account the possibility that the flat-field lamps intensity may drift slightly with time; i.e., a good exposure sequence might be 1sec, 20 sec, 2 sec, 2 sec, 20 sec, 1 sec. The **findshutcorr** routine currently lives in the **nlocal** package.

I.5. Checking the weather without leaving the dome: *wdisplay* and picking up the latest forecast.

Are you wondering how likely it is going to remain clear, or are you considering packing it in for the night? The **wdisplay** task in the **nlocal** package may help you decide what to do. Three times an hour, 24 hours a day, a new GOES weather satellite picture in one of three wavelength bands (visible, IR, and a water-vapor narrow-IR band) is downloaded. It is fun, easy, and informative to display the last four images, and quickly blink between to see what's coming your way, or leaving you alone. Simply run

```
cl> wdisplay ir four+
```

to load the last four images into your ximtool window. (You may have to select the menu item "FitFrame".) Hit CNTL-F repeatedly to make a little movie. Further information can be gleaned from the help page.

If you would like to compare your guessimate with that of an expert, you can also easily retrieve the most recent weather prediction. Put the mouse in a blank area of the screen and select ``Weather Forecast'' under ``Astronomer Tools''. You'll be presented with a map of the US with little dots on it; place the cursor near Tucson, and click. You can check the weather back home the same way, of course.

Appendix J: Other Nifty Things

There is a prototype automatic logging routine that constructs a TeX version of the observing log. Instructions may be obtained via anonymous ftp to ``ftp.noao.edu'', cd kpno/manuals, binary, get ccdlogs2.ps.Z. Comments should be directed to tlauer@noao.edu.

Under ``Astronomer tools'' in the root menu, there is an ephemeris program (``xephem''), provided by Elwood Downey, that will tell you the Julian date and times of sunrise, sunset, and twilight, and many more wonderful things. (Do a Unix level ``man xephem'' for more information.)

Appendix Z: Revision History

```
Version Date    Author  Comment
------- ----    ------  -------

0.0     ???     CTIO    original CTIO Arcon User's Manual

0.1     6/1/95  Kennedy programmer pass 1

0.2     6/15/95 Kennedy programmer pass 2

0.3     7/2/95  Silva   drs pass 1 - shipped to Armandroff

0.4     7/12/95 Silva   drs pass 2 - incorp Armandroff comments,
                                     shipped to Massey

0.5     7/21/95 Silva   incorp Massey comments, issued as beta 0.5

0.6     8/2/95  Silva   incorp von Hippel comments, fixed HTML refs,
                        added new appendices with misc IRAF
                        information, issued as beta 0.6

TODO:

(1) Add "focus" command information when available (mid-September).

(2) Beef up HTML internal referencing

(3) Add "active" Table of Contents at top of document

(4) Add "obsinit" information when it becomes available

(5) Investigate/document how to write observing scripts (if possible).

(6) Investigate/document Arcon/CLI interface (if possible).

(7) Add xtapemon stuff.

(8) Re-review ICE manual for other tidbits to include.
```